CLAIMS

What is claimed is:

1. A system comprising:

one or more processors; and

memory coupled to the processor, the memory containing one or more sequences of instructions for establishing sessions between a client and a server over a communications network, wherein execution of the one or more sequences of instructions by the one or more processors causes the processors to perform:

receiving a first request to establish a first session between a client and a first server, wherein the request includes user identification information;

determining, based on the user identification information, whether the first session between the client and the first server should be established, and if so,

authorizing the first session between the client and the first server, and

causing the user identification information to be stored in a cache; and

authorizing a second session between the client and the first server in response to a second request for the second session, based on the user identification information from the first request that is stored in the cache.

2. The system as recited in claim 1, wherein the user identification information includes a username and a one-time password (OTP), and wherein authorizing a second session comprises determining whether the OTP is valid at the time that the second request is received.

3. The system as recited in claim 2, wherein authorizing a second session comprises determining whether the username and the OTP are in the cache, and if the username and the OTP are not in the cache, generating a request that can be sent to a password server to determine whether the OTP is currently valid.

4. The system as recited in claim 2, wherein authorizing a second session comprises determining whether the username and the OTP are in the cache, and if the username and the OTP are not in the cache, determining whether the username and the OTP are still valid.

5. The system as recited in claim 4, wherein determining whether the username and the OTP are still valid comprises:

creating and storing a cached time value for the username and the OTP that indicates how long the username and the OTP have been stored in the cache; and

comparing the cached time value with an expiration time-out value to determine whether the username and OTP are still valid.

6. The system as recited in claim 4, wherein the step of determining whether the username and the OTP are still valid comprises determining whether an active

session currently exists between the client and the first server at the time of the second request.

7. The system as recited in claim 1, wherein the user identification information includes a username and a one-time password (OTP), and wherein authorizing the second session comprises:

generating instructions to a second server for determining whether the username and OTP are currently in a second cache of the second server; and

generating a request to a password server to authenticate the OTP; and

generating instructions to the second server for caching the username and the OTP in memory at the second server.

8. The system as recited in claim 1, wherein authorizing a second session comprises:

receiving the second request from the first server, wherein the second request includes user identification information that contains a username and a second one-time password (OTP);

determining whether the username and the second OTP correspond to user identification information stored in the cache, and if so, authorizing the second session between the client and the first server.

9. The system as recited in claim 1, wherein receiving a first request to establish a first session between the client and the first server comprises receiving a first request in a Challenge Handshake Authentication Protocol, and wherein the sequences of instructions cause the one or more processors to perform:

validating the client using the Challenge Handshake Authentication Protocol before authorizing the first session between the client and the first server.

10. The system as recited in claim 1, wherein receiving a first request to establish a first session between the client and the first server comprises receiving the first request in a Password Authentication Protocol, and wherein the sequences of instructions cause the one or more processors to perform:
validating the client using the Password Authentication Protocol before authorizing the first session between the client and the first server.

11. The system as recited in claim 1, wherein receiving a second request to establish a second session between the client and the first server comprises receiving the second request in a Challenge Handshake Authentication Protocol, and wherein the sequences of instructions cause the one or more processors to perform:
validating the client using the Challenge Handshake Authentication Protocol before authorizing the second session between the client and the first server.

12. The system as recited in claim 1, wherein receiving a second request to establish a second session between the client and the first server comprises receiving the second request based a Password Authentication Protocol, and wherein the sequences of instructions cause the one or more processors to perform:
validating the client using the Password Authentication Protocol before authorizing the second session between the client and the first server.

13. The system as recited in claim 1, wherein receiving a first request comprises receiving a one-time password that is generated by a Token card.

14. The system as recited in claim 13, wherein receiving a second request comprises receiving the same one-time password as received in the first request.

15. The system as recited in claim 1, wherein the sequences of instructions cause the one or more processors to perform:

establishing a first Point-to-Point (PPP) session between the client and the first server.

16. The system as recited in claim 1, wherein the sequences of instructions cause the one or more processors to perform:

establishing a first Serial Line Internet Protocol (SLIP) session between the client and the first server.

17. The system as recited in claim 1, wherein the sequences of instructions cause the one or more processors to perform:

establishing a first second Point-to-Point (PPP) session between the client and the first server.

18. The system as recited in claim 1, wherein the sequences of instructions cause the one or more processors to perform:

establishing a first second Serial Line Internet Protocol (SLIP) session between the client and the first server.

19. The system as recited in claim 1,

wherein the first request includes a first username and a first one-time password (OTP) and the second request includes a second username and a second one-time password (OTP);

wherein storing the user identification information comprises storing the first username and the first OTP in a cache; and

wherein authorizing the second session comprises determining that the second OTP corresponds to the first OTP that is in the cache.

20. The system recited in claim1, wherein the sequences of instructions cause the one or more processors to perform:

identifying, based on a username from the user identification information, a set of access rights that is used by the first server in determining what may be performed by a user during the first session; and

transmitting the set of access rights to the first server.

21. A method comprising computer-implemented steps of:

determining, based on user identification information that is included in a first request to establish a first session between a client and a first server, whether the first session between the client and the first server should be established, and if so,

authorizing the first session between the client and the first server, and

causing the user identification information to be stored in a cache; and

authorizing a second session between the client and the first server in response to a second request for the second session, based on the user identification information from the first request that is stored in the cache.

22. The method recited in claim 21, wherein the first server is a network access server and the steps of determining, authorizing the first session, causing, and authorizing the second session are performed by an Authorization, Authentication, and Accounting server.

23. The method recited in claim 21, wherein the second request includes user identification information that contains a username and a second one-time password (OTP), and wherein authorizing a second session comprises:

determining whether the username and the second OTP correspond to user identification information stored in the cache, and if so, authorizing the second session between the client and the first server.

24. The method recited in claim 23, further comprising the computer-implemented step of receiving the second request from the first server.

25. The method recited in 21, wherein the user identification information includes a username and a one-time password (OTP), and wherein authorizing a second session comprises determining whether the OTP is valid at the time that the second request is received.

26. The method recited in claim 25, wherein authorizing a second session comprises determining whether the username and the OTP are in the cache, and if the username and the OTP are not in the cache, generating a request that can be sent to a password server to determine whether the OTP is currently valid.

27. The method recited in claim 21, wherein the user identification information includes a username and a one-time password (OTP), and wherein authorizing the second session comprises:

generating instructions to a second server for determining whether the username and OTP are currently in a second cache of the second server; and

generating a request to a password server to authenticate the OTP; and

generating instructions to the second server for caching the username and the OTP in memory at the second server.

28. The method recited in claim 27, wherein the second server is an Authorization, Authentication, and Accounting server.

29. The method recited in claim 21, further comprising the computer-implemented steps of:

identifying, based on a username from the user identification information, a set of access rights that is used by the first server in determining what may be performed by a user during the first session; and

transmitting the set of access rights to the first server.

30. A method comprising computer-implemented steps of:

receiving at a first server a first request to establish a first session between a client and the first server, wherein the first request includes first user identification information;

passing at least the first user identification information to a second server for use by the second server in determining whether the first session between the

client and the first server should be established, and if so, authorizing the first session between the client and the first server, and for storing in a cache at the second server;

receiving at the first server a second request to establish a second session between the client and the first server, wherein the second request includes second user identification information;

passing at least the second user identification information to the second server for use by the second server in determining, based on the first user identification information that is stored in the cache and on the second user identification information, whether the second session between the client and the first server should be established, and if so, authorizing the second session between the client and the first server.

31. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, cause the one or more processors to perform at least the steps of:

determining, based on user identification information that is included in a first request to establish a first session between a client and a first server, whether the first session between the client and the first server should be established, and if so,

authorizing the first session between the client and the first server, and

causing the user identification information to be stored in a cache; and

authorizing a second session between the client and the first server in response to a second request for the second session, based on the user identification information from the first request that is stored in the cache.

32. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, cause the one or more processors to at least the steps of:

receiving at a first server a first request to establish a first session between a client and the first server, wherein the first request includes first user identification information;

passing at least the first user identification information to a second server for use by the second server in determining whether the first session between the client and the first server should be established, and if so, authorizing the first session between the client and the first server, and for storing in a cache at the second server;

receiving at the first server a second request to establish a second session between the client and the first server, wherein the second request includes second user identification information;

passing at least the second user identification information to the second server for use by the second server in determining, based on the first user identification information that is stored in the cache and on the second user identification information, whether the second session between the client and the first server should be established, and if so, authorizing the second session between the client and the first server.

33. A system comprising:

one or more processors; and

memory coupled to the processor, the memory containing one or more sequences of instructions which, when executed by the one or more processors cause the processors to at least the steps of:

determining, based on user identification information that is included in a first request to establish a first session between a client and a first server, whether the first session between the client and the first server should be established, and if so,

authorizing the first session between the client and the first server, and

causing the user identification information to be stored in a cache; and

authorizing a second session between the client and the first server in response to a second request for the second session, based on the user identification information from the first request that is stored in the cache.

34. The system recited in claim 33, wherein the first server is a network access server and the system comprises an Authorization, Authentication, and Accounting server.

35. The system recited in claim 33, wherein the second request includes user identification information that contains a username and a second one-time password (OTP), and wherein authorizing a second session comprises:

determining whether the username and the second OTP correspond to user identification information stored in the cache, and if so, authorizing the second session between the client and the first server.

36. The system recited in claim 35, wherein the instructions cause the one or more processors to perform at least the step of receiving the second request from the first server.

37. The system recited in claim 33, wherein the user identification information includes a username and a one-time password (OTP), and wherein authorizing a second session comprises determining whether the OTP is valid at the time that the second request is received.

38. The system recited in claim 37, wherein authorizing a second session comprises determining whether the username and the OTP are in the cache, and if the username and the OTP are not in the cache, generating a request that can be sent to a password server to determine whether the OTP is currently valid.

39. The system recited in claim 33, wherein the user identification information includes a username and a one-time password (OTP), and wherein authorizing the second session comprises:

generating instructions to a second server for determining whether the username and OTP are currently in a second cache of the second server; and

generating a request to a password server to authenticate the OTP; and

generating instructions to the second server for caching the username and the OTP in memory at the second server.

40. The system recited in claim 39, wherein the second server is an Authorization, Authentication, and Accounting server.

41. The system recited in claim 33, wherein the instructions cause the one or more processors to perform at least the steps of:

identifying, based on a username from the user identification information, a set of access rights that is used by the first server in determining what may be performed by a user during the first session; and

transmitting the set of access rights to the first server.

42. A system comprising:

one or more processors; and

memory coupled to the processor, the memory containing one or more sequences of instructions which, when executed by the one or more processors cause the processors to at least the steps of:

receiving at a first server a first request to establish a first session between a client and the first server, wherein the first request includes first user identification information;

passing at least the first user identification information to a second server for use by the second server in determining whether the first session between the client and the first server should be established, and if so, authorizing the first session between the client and the first server, and for storing in a cache at the second server;

receiving at the first server a second request to establish a second session between the client and the first server, wherein the second request includes second user identification information;

passing at least the second user identification information to the second server for use by the second server in determining, based on the first user identification information that is stored in the cache and on the second user identification information, whether the second session between the client and the first server should be established, and if so, authorizing the second session between the client and the first server.